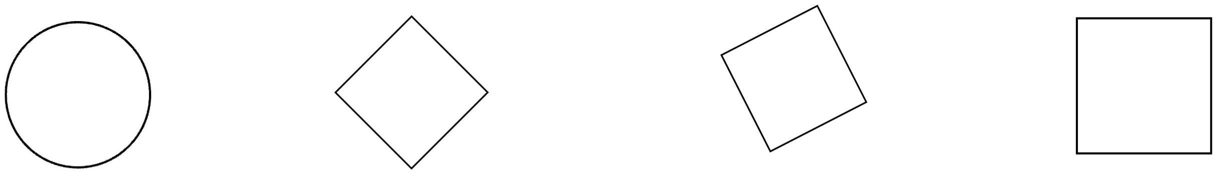
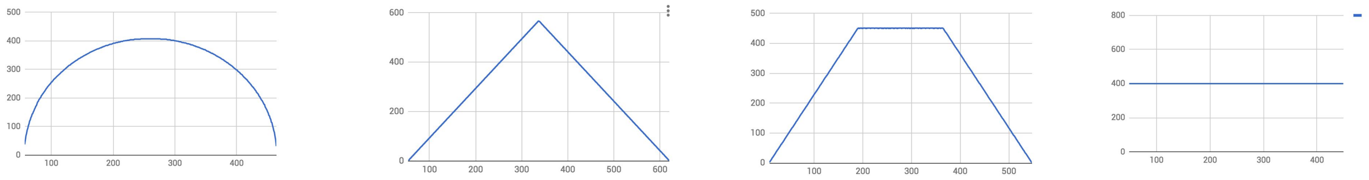
ShapeFinder

The program:

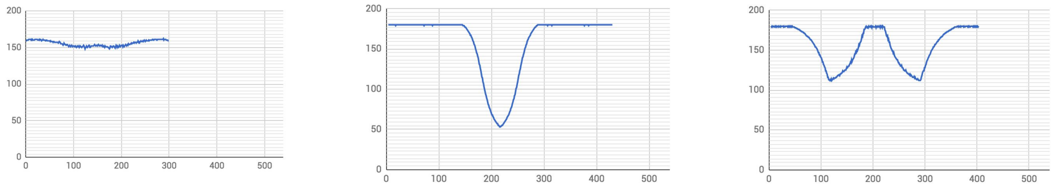
Shape finder, as indicated in the name of the program, can distinguish two kind of shapes: circles and squares.



The shape must be black on a white background, then the program iterates over it, from top to bottom, making a curve of the « evolution of width » of the shape.



Then it iterates over the curve, placing three points at equal distance on it, and makes another curve with the evolution of the angle formed by this three points.



EXCEPTION DETECTED => SQUARE

Finally, for each curve based on the average of all points, we define a threshold that help us, when reached, to identify the « spikes » you can see on the above curves that are corresponding to sharp angles on the shape. Then with this method the program can compare features extracted from images and tell the difference between circle and square.

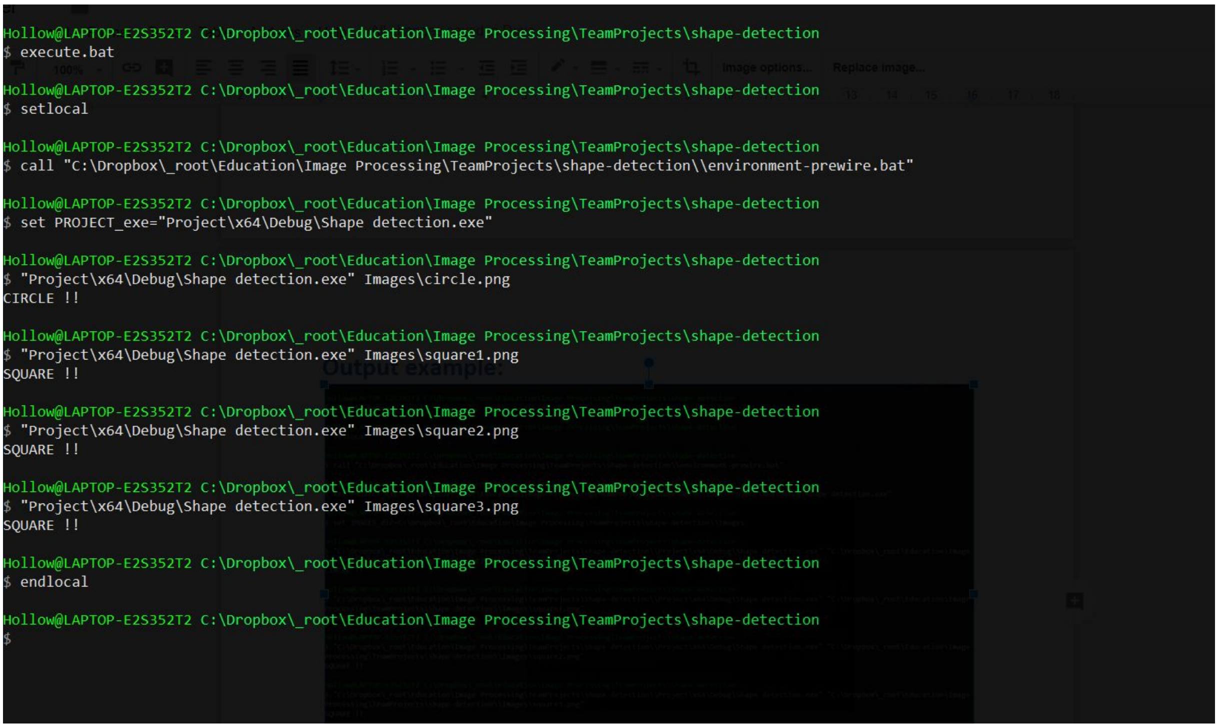
The team:

We were two working on this project and it took us about an afternoon and one meeting to develop this solution: we used git as versioning system to work on the same code. It took us another evening to set up the project to work out-of-the-box on a windows machine.

Dylan Heirstraeten: “It was an interesting project and it introduced the difficulty of patterns recognition in pictures. I think we could have gone further with multiplying the features generators or by adding some pre-processing but our solution is sufficient for the problem asked.”

Sebastianas Malinauskas: “The project presented a challenge to a seemingly simple problem. After iterating through a few ideas, we finally got to a solution that got the job done. We spent a lot of time trying to make OpenCV work on C++. This could be avoided by making future OpenCV projects be based on Python.”

Output example:



```
Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ execute.bat

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ setlocal

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ call "C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection\\environment-prewire.bat"

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ set PROJECT_exe="Project\x64\Debug\Shape detection.exe"

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ "Project\x64\Debug\Shape detection.exe" Images\circle.png
CIRCLE !!

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ "Project\x64\Debug\Shape detection.exe" Images\square1.png
SQUARE !!

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ "Project\x64\Debug\Shape detection.exe" Images\square2.png
SQUARE !!

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ "Project\x64\Debug\Shape detection.exe" Images\square3.png
SQUARE !!

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$ endlocal

Hollow@LAPTOP-E2S352T2 C:\Dropbox\_root\Education\Image Processing\TeamProjects\shape-detection
$
```

Projection information:

Dependencies:
    OpenCV 2.4.13 x64
    Visual Studio 2017

In order to ensure that the correct version of OpenCV is used with the project, we have bundled the OpenCV prebuilt libraries and added a few batch files to make things easier to execute:

1) **"launchSolution.bat": Setups the environment to use the pre-bundled OpenCV libraries and then launches the Visual Studio solution.**
2) **"execute.bat": Setups the environment to use the pre-bundled OpenCV libraries and then launches the project executbable with different images located in Images\ folder as an argument.**